

ЛАБОРАТОРНА РОБОТА № 2

Породжувальні шаблони

Мета: навчитися реалізовувати породжувальні шаблони проєктування.

Хід роботи:

Завдання 0. Підготовка до виконання завдання

1. Створити директорію або гілку lab-2 в Вашому репозиторії з лабораторками.
2. Перейти до Завдання 1 😊

Завдання 1. Фабричний метод.

1. Напишіть систему класів для реалізації функціоналу створення різних типів підписок для відео провайдера.
2. Кожна з підписок повинна мати щомісячну плату, мінімальний період підписки та список каналів й інших можливостей.
3. Види підписок: DomesticSubscription, EducationalSubscription, PremiumSubscription.
4. Придбати (тобто створити) підписку можна за допомогою трьох різних класів: WebSite, MobileApp, ManagerCall, кожен з них має реалізувати свою логіку створення підписок.
5. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
6. Підготуйте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

Лістинг програми:

```
using Lab2.Task1.Subscriptions;

namespace Lab2.Task1.Factories;

public interface IFactory
{
    public ISubscription CreateSubscription(string country, int months, string? studentId);
}
```

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».23.121.09.000 – Пр2		
Розроб.	Геращенко Д. І.				Звіт з лабораторної роботи		
Перевір.	Фант М. О.						
Керівник							
Н. контр.							
Зав. каф.					ФІКТ Гр. ІПЗ-23-1[2]		
					Літ.	Арк.	Аркушів
						1	19

```

using Lab2.Task1.Subscriptions;

namespace Lab2.Task1.Factories;

public class ManagerCall : IFactory
{
    public ISubscription CreateSubscription(string country, int months, string? studentId)
    {
        // We don't want to bother with the students on the phone
        return (country == "US") ? new DomesticSubscription() : new PremiumSubscription();
    }
}
using Lab2.Task1.Subscriptions;

namespace Lab2.Task1.Factories;

public class MobileApp : IFactory
{
    public ISubscription CreateSubscription(string country, int months, string? studentId)
    {
        // Suppose that Apple forbids discrimination by the country of origin xD
        return (studentId != null) ? new EducationalSubscription() : new PremiumSubscription();
    }
}
using Lab2.Task1.Subscriptions;

namespace Lab2.Task1.Factories;

public class WebSite : IFactory
{
    public ISubscription CreateSubscription(string country, int months, string? studentId)
    {
        if (country == "US")
        {
            return new DomesticSubscription();
        }
        if (studentId != null)
        {
            return new EducationalSubscription();
        }
        else
        {

```

Змн.	Арк.	Геращенко Д. І.		
		Фант М. О.		

```

        return new PremiumSubscription();
    }
}
}

namespace Lab2.Task1.Subscriptions;

public interface ISubscription
{
    public string Name { get; }
    public decimal MonthlyFee { get; }
    public int MinMonthDuration { get; }
    public List<string> Channels { get; }
    public List<string> Features { get; }
}

namespace Lab2.Task1.Subscriptions;

public class DomesticSubscription : ISubscription
{
    public string Name => "Domestic";
    public decimal MonthlyFee => 9.99m;
    public int MinMonthDuration => 3;
    public List<string> Channels => ["BBC", "CNN", "Discovery"];
    public List<string> Features => ["HD", "Opus Audio"];
}

namespace Lab2.Task1.Subscriptions;

public class EducationalSubscription : ISubscription
{
    public string Name => "Educational";
    public decimal MonthlyFee => 4.99m;
    public int MinMonthDuration => 12;
    public List<string> Channels => ["Discovery"];
    public List<string> Features => ["HD"];
}

namespace Lab2.Task1.Subscriptions;

public class PremiumSubscription : ISubscription
{
    public string Name => "Premium";
    public decimal MonthlyFee => 19.99m;
    public int MinMonthDuration => 1;
    public List<string> Channels => ["BBC", "CNN", "Discovery", "Fox", "NBC"];
    public List<string> Features => ["HD", "4K", "Opus Audio", "Dolby Atmos"];
}

using Lab2.Task1.Factories;
using Lab2.Task1.Subscriptions;
using Xunit;

```

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

namespace Lab2.Task1.Tests;

public class FactoriesTests
{
    [Fact]
    public void ManagerCall()
    {
        IFactory factory = new ManagerCall();
        Assert.Equal("Premium", CreateSubscription(factory, "UA", 12, null).Name);
        Assert.Equal("Domestic", CreateSubscription(factory, "US", 12,
"IPZ231").Name);
    }

    [Fact]
    public void MobileApp()
    {
        IFactory factory = new MobileApp();
        Assert.Equal("Premium", CreateSubscription(factory, "US", 12, null).Name);
        Assert.Equal("Educational", CreateSubscription(factory, "UA", 12,
"IPZ231").Name);
    }

    [Fact]
    public void WebSite()
    {
        IFactory factory = new WebSite();
        Assert.Equal("Premium", CreateSubscription(factory, "UA", 12, null).Name);
        Assert.Equal("Educational", CreateSubscription(factory, "UA", 12,
"IPZ231").Name);
        Assert.Equal("Domestic", CreateSubscription(factory, "US", 12,
"IPZ231").Name);
    }

    public ISubscription CreateSubscription(IFactory factory, string country, int
months, string? studentId)
    {
        return factory.CreateSubscription(country, months, studentId);
    }
}

```

Результат виконання програми:

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

    ✓ Lab2 (net9.0) 78ms
    ✓ Lab2.Task1.Tests 17ms
    ✓ FactoriesTests 17ms
      ✓ ManagerCall 3.0ms
      ✓ MobileApp 5.0ms
      ✓ WebSite 9.0ms

```

Рис 1. Результат виконання завдання 1

Блок схема:

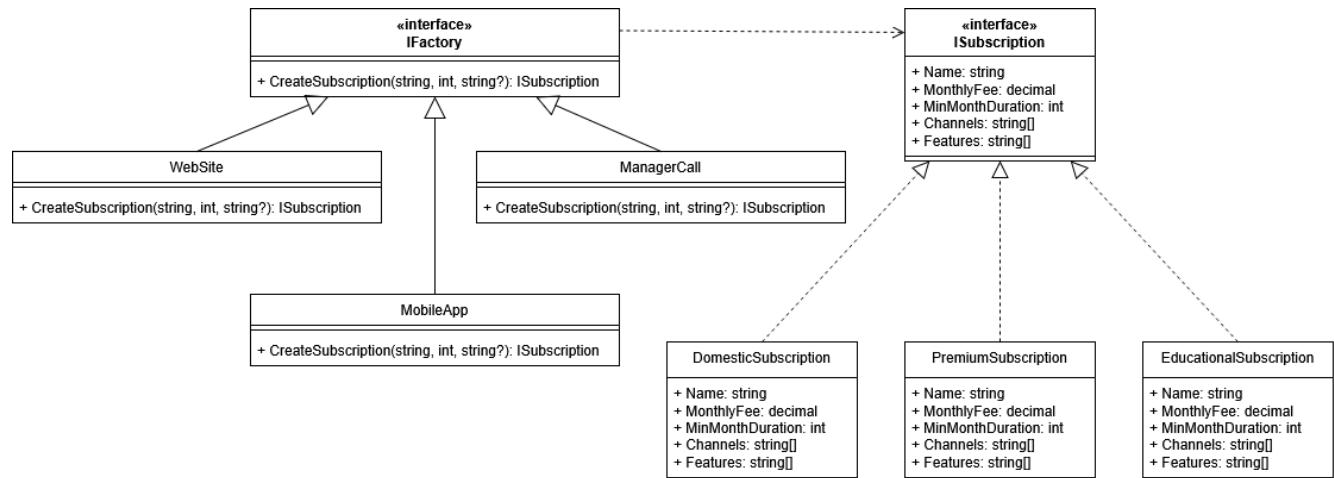


Рис 2. UML діаграма до завдання 1

Завдання 2. Абстрактна фабрика

- Створіть фабрику виробництва техніки.
- На фабриці мають створюватися різні девайси (наприклад, Laptop, Netbook, EBook, Smartphone) для різних брендів (IPhone, Kiaomi, Galaxy).
- Покажіть правильність роботи свого коду запустивши його в головному методі програми.
- Підгответте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

Лістинг програми:

```

namespace Lab2.Task2.Devices;

public interface IEBook {
    public string Screen { get; }
}

namespace Lab2.Task2.Devices;

public interface ILaptop {
    public string CPU { get; }
    public string GPU { get; }
}

```

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

}

namespace Lab2.Task2.Devices;

public interface INetbook {
    public string Storage { get; }
}
namespace Lab2.Task2.Devices;

public interface ISmartphone {
    public string CPU { get; }
}
using Lab2.Task2.Devices;

namespace Lab2.Task2.Factories;

public interface IFactory
{
    public ILaptop CreateLaptop();
    public INetbook CreateNetbook();
    public IEBook CreateEBook();
    public ISmartphone CreateSmartphone();
}

using Lab2.Task2.Devices;

namespace Lab2.Task2.Factories;

public class GalaxyFactory : IFactory
{
    public ILaptop CreateLaptop() => new GalaxyLaptop();
    public INetbook CreateNetbook() => new GalaxyNetbook();
    public IEBook CreateEBook() => new GalaxyEBook();
    public ISmartphone CreateSmartphone() => new GalaxySmartphone();
}

public class GalaxyLaptop : ILaptop
{
    public string CPU => "i5-8250U";
    public string GPU => "GTX 1050";
}

public class GalaxyNetbook : INetbook
{
    public string Storage => "128 GB";
}

public class GalaxyEBook : IEBook
{
}

```

		Геращенко Д. І.			Арк.
		Фаніт М. О.			
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».23.121.09.000 – Пр2

```

        public string Screen => "AMOLED";
    }

public class GalaxySmartphone : ISmartphone
{
    public string CPU => "Exynos";
}

using Lab2.Task2.Devices;

namespace Lab2.Task2.Factories;

public class IPhoneFactory: IFactory
{
    public ILaptop CreateLaptop() => new IPhoneLaptop();
    public INetbook CreateNetbook() => new IPhoneNetbook();
    public IEBook CreateEBook() => new IPhoneEBook();
    public ISmartphone CreateSmartphone() => new IPhoneSmartphone();
}

public class IPhoneLaptop : ILaptop
{
    public string CPU => "M2 Max";
    public string GPU => "M2 Max";
}

public class IPhoneNetbook : INetbook
{
    public string Storage => "32 GB";
}

public class IPhoneEBook : IEBook
{
    public string Screen => "OLED 4K";
}

public class IPhoneSmartphone : ISmartphone
{
    public string CPU => "A12";
}

using Lab2.Task2.Devices;

namespace Lab2.Task2.Factories;

public class XiaomiFactory: IFactory
{
    public ILaptop CreateLaptop() => new XiaomiLaptop();
    public INetbook CreateNetbook() => new XiaomiNetbook();
}

```

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фаніт М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

```

        public IEBook CreateEBook() => new KiaomiEBook();
        public ISmartphone CreateSmartphone() => new KiaomiSmartphone();
    }

    public class KiaomiLaptop : ILaptop
    {
        public string CPU => "Ryzen 5 3900";
        public string GPU => "Vega 8";
    }

    public class KiaomiNetbook : INetbook
    {
        public string Storage => "256 GB";
    }

    public class KiaomiEBook : IEBook
    {
        public string Screen => "IPC 1080p";
    }

    public class KiaomiSmartphone : ISmartphone
    {
        public string CPU => "Snapdragon 860";
    }

using Lab2.Task2.Devices;
using Lab2.Task2.Factories;
using Xunit;

namespace Lab2.Task2.Tests;

public class FactoriesTests
{
    [Fact]
    public void IPhoneDevices()
    {
        IFactory factory = new IPhoneFactory();
        var (laptop, netbook) = CreateDesktops(factory);
        Assert.Equal("M2 Max", laptop.CPU);
        Assert.Equal("M2 Max", laptop.GPU);
        Assert.Equal("32 GB", netbook.Storage);

        var (ebook, smartphone) = CreateHandhelds(factory);
        Assert.Equal("OLED 4K", ebook.Screen);
        Assert.Equal("A12", smartphone.CPU);
    }

    [Fact]
    public void KiaomiDevices()
}

```

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

```

{
    IFactory factory = new XiaomiFactory();
    var (laptop, netbook) = CreateDesktops(factory);
    Assert.Equal("Ryzen 5 3900", laptop.CPU);
    Assert.Equal("Vega 8", laptop.GPU);
    Assert.Equal("256 GB", netbook.Storage);

    var (ebook, smartphone) = CreateHandhelds(factory);
    Assert.Equal("IPC 1080p", ebook.Screen);
    Assert.Equal("Snapdragon 860", smartphone.CPU);
}

[Fact]
public void GalaxyDevices()
{
    IFactory factory = new GalaxyFactory();
    var (laptop, netbook) = CreateDesktops(factory);
    Assert.Equal("i5-8250U", laptop.CPU);
    Assert.Equal("GTX 1050", laptop.GPU);
    Assert.Equal("128 GB", netbook.Storage);

    var (ebook, smartphone) = CreateHandhelds(factory);
    Assert.Equal("AMOLED", ebook.Screen);
    Assert.Equal("Exynos", smartphone.CPU);
}

public (ILaptop, INetbook) CreateDesktops(IFactory factory)
{
    return (factory.CreateLaptop(), factory.CreateNetbook());
}

public (IEBook, ISmartphone) CreateHandhelds(IFactory factory)
{
    return (factory.CreateEBook(), factory.CreateSmartphone());
}
}

```

Результат виконання програми:



Рис 3. Результат виконання завдання 2

Блок схема:

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

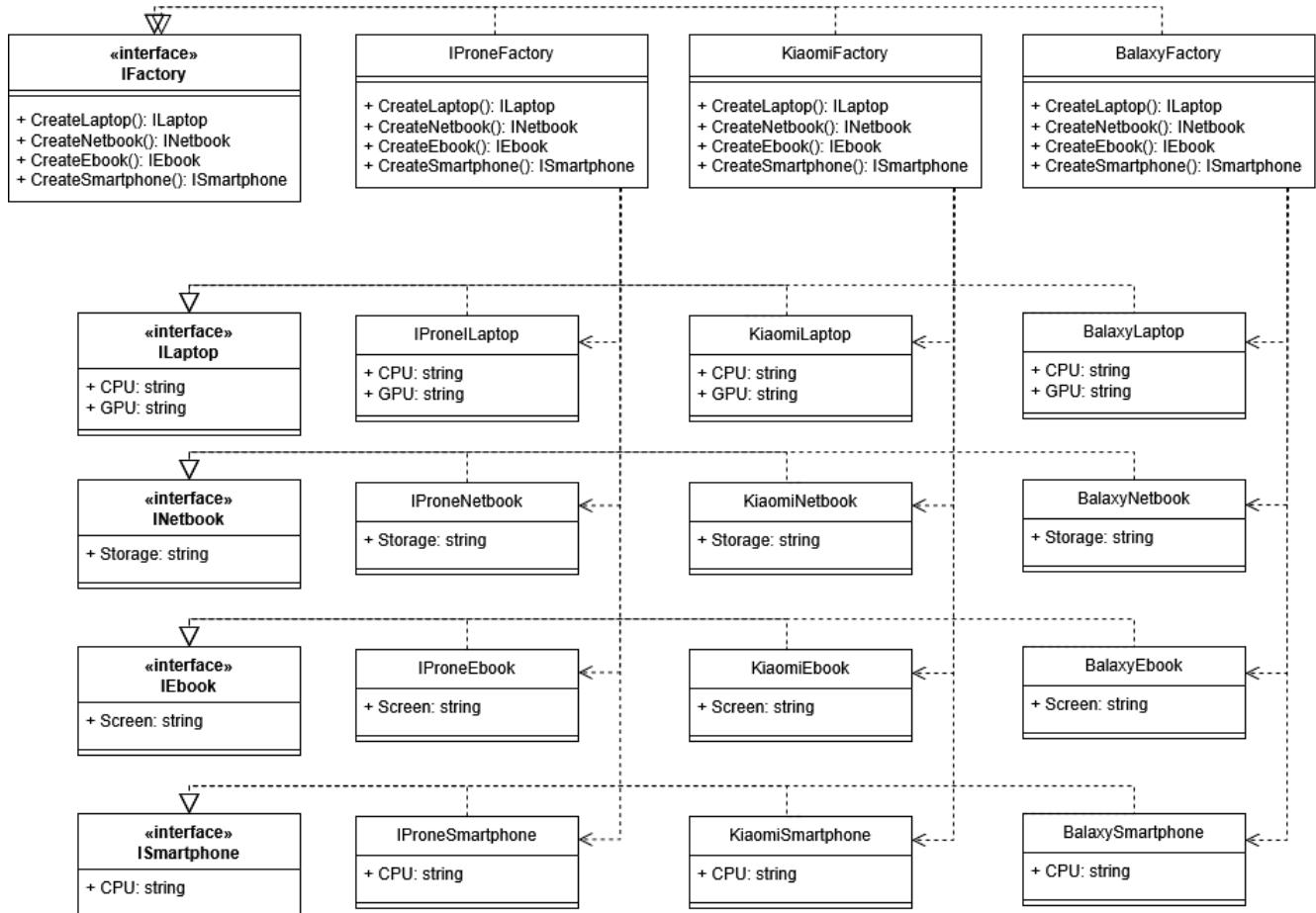


Рис 4. UML діаграма до завдання 2

Завдання 3. Одинак.

- Створіть клас Authenticator таким чином, щоб бути впевненим, що цей клас може створити лише один екземпляр, незалежно від кількості потоків і класів, що його наслідують.
- Покажіть правильність роботи свого коду запустивши його в головному методі програми

Лістинг програми:

```

namespace Lab2.Task3;

public sealed class Authenticator
{
    private static Authenticator? _instance;
    private static readonly object _lock = new object();

    private Authenticator() { }

    public static Authenticator GetInstance()
    {
        if (_instance == null)
        {
            lock (_lock)
            {

```

Геращенко Д. І.								
Фант М. О.								
Змн.	Арк.	№ докум.	Підпис	Дата		ДУ «Житомирська політехніка».23.121.09.000 – Пр2		Арк. 10

```

        if (_instance == null)
        {
            _instance = new Authenticator();
        }
    }
    return _instance;
}

public bool Login(string username, string password)
{
    return username == "admin" && password == "admin";
}
}
using Lab2.Task3;
using Xunit;

namespace Lab2.Task3.Tests;

public class SingletonTests
{
    [Fact]
    public async Task AuthenticatorMultithreading()
    {
        var tasks = new List<Task<Authenticator>>();
        for (int i = 0; i < 100; i++)
        {
            tasks.Add(Task.Run(() => Authenticator.GetInstance()));
        }

        var first = await tasks.First();
        foreach (var instance in tasks)
        {
            Assert.Same(first, await instance);
        }
    }
}

```

Результат виконання програми:



Рис 5. Результат виконання завдання 3

Завдання 4. Прототип.

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Пр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

- Створіть клас Virus. Він повинен містити вагу, вік, ім'я, вид і масив дітей, екземплярів Virus.
- Створіть екземпляри для цілого “сімейства” вірусів (мінімум три покоління).
- За допомогою шаблону Прототип реалізуйте можливість клонування наявних вірусів.
- При клонуванні віруса-батька повинні клонуватися всі його діти.
- Покажіть правильність роботи свого коду запустивши його в головному методі програми.

Лістинг програми:

```

namespace Lab2.Task4;

public interface IPrototype<T>
{
    T Clone();
}

namespace Lab2.Task4;

public class Virus(double weight, int age, string name, string kind) :
    IPrototype<Virus>
{
    public double Weight { get; set; } = weight;
    public int Age { get; set; } = age;
    public string Name { get; set; } = name;
    public string Kind { get; set; } = kind;
    public List<Virus> Children { get; set; } = new List<Virus>();

    public Virus Clone()
    {
        var virus = new Virus(Weight, Age, Name, Kind);
        virus.Children = new List<Virus>();
        foreach (var child in Children)
        {
            virus.Children.Add(child.Clone());
        }
        return virus;
    }
}

using Lab2.Task4;
using Xunit;

namespace Lab2.Task4.Tests;

public class PrototypeTests
{
    [Fact]

```

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

```

public void VirusCloning()
{
    var gen1_1 = new Virus(9, 50, "Virus A1", "Alpha");
    var gen2_1 = new Virus(5, 25, "Virus B1", "Beta");
    var gen2_2 = new Virus(6, 28, "Virus B2", "Beta");
    gen1_1.Children.Add(gen2_1);
    gen1_1.Children.Add(gen2_2);

    var gen3_1 = new Virus(1, 5, "Virus C1", "Gamma");
    var gen3_2 = new Virus(2, 8, "Virus C2", "Gamma");
    gen2_1.Children.Add(gen3_1);
    gen2_1.Children.Add(gen3_2);

    var gen1_2 = gen1_1.Clone();
    Assert.NotSame(gen1_1, gen1_2);
    Assert.Equal(gen1_1.Weight, gen1_2.Weight);
    Assert.Equal(gen1_1.Age, gen1_2.Age);
    Assert.Equal(gen1_1.Name, gen1_2.Name);
    Assert.Equal(gen1_1.Kind, gen1_2.Kind);

    Assert.NotSame(gen1_1.Children[0], gen1_2.Children[0]);
    Assert.NotSame(gen1_1.Children[1], gen1_2.Children[1]);

    Assert.Equal(gen1_1.Children[0].Weight, gen1_2.Children[0].Weight);
    Assert.Equal(gen1_1.Children[0].Age, gen1_2.Children[0].Age);
    Assert.Equal(gen1_1.Children[0].Name, gen1_2.Children[0].Name);
    Assert.Equal(gen1_1.Children[0].Kind, gen1_2.Children[0].Kind);

    Assert.Equal(gen1_1.Children[1].Weight, gen1_2.Children[1].Weight);
    Assert.Equal(gen1_1.Children[1].Age, gen1_2.Children[1].Age);
    Assert.Equal(gen1_1.Children[1].Name, gen1_2.Children[1].Name);
    Assert.Equal(gen1_1.Children[1].Kind, gen1_2.Children[1].Kind);
}

}

```

Результат виконання програми:

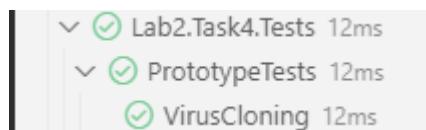


Рис 6. Результат виконання завдання 4

Завдання 5. Будівельник.

- Створіть клас HeroBuilder, який буде створювати персонажа гри, поступово додаючи до нього різні ознаки, наприклад зріст, статуру,

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Пр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

колір волосся, очей, одяг, інвентар тощо (можете включити фантазію).

2. Створіть клас EnemyBuilder, який буде реалізовувати єдиний інтерфейс з HeroBuilder. Відмінністю між ними можуть бути спеціальні методи для творення добра або зла, а також списки добрих і злих справ відповідно.
3. За допомогою свого білдера і класу-директора створіть героя (або геройню) своєї мрії 😊, а також свого найзапеклішого ворога.
4. Зверніть увагу, що Ваші білдери повинні реалізовувати текучий інтерфейс (fluent interface).
5. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
6. Підгответте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

Лістинг програми:

```
namespace Lab2.Task5;

public class Character
{
    public string Name { get; set; } = "Default";
    public int Height { get; set; } = 170;
    public string Build { get; set; } = "Normal";
    public string HairColor { get; set; } = "Black";
    public string EyeColor { get; set; } = "Black";
    public string Clothing { get; set; } = "None";
    public List<string> Inventory { get; set; } = [];
    public string Alignment { get; set; } = "Neutral";

    public override string ToString()
    {
        return $"""
            Name: {Name}
            Height: {Height}
            Build: {Build}
            HairColor: {HairColor}
            EyeColor: {EyeColor}
            Clothing: {Clothing}
            Inventory: {Inventory}
            Alignment: {Alignment}
            """;
    }
}

namespace Lab2.Task5.BUILDERS;
public interface ICharacterBuilder
{
```

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

```

void Reset();
ICharacterBuilder SetName(string name);
ICharacterBuilder SetHeight(int height);
ICharacterBuilder SetBuild(string build);
ICharacterBuilder SetHairColor(string hairColor);
ICharacterBuilder SetEyeColor(string eyeColor);
ICharacterBuilder SetClothing(string clothing);
ICharacterBuilder AddInventory(string inventory);
Character Build();
}

namespace Lab2.Task5.BUILDERS;

public class EnemyBuilder : ICharacterBuilder
{
    private Character _character = new Character();

    public EnemyBuilder()
    {
        Reset();
    }

    public void Reset()
    {
        _character = new Character();
        _character.Alignment = "Evil";
    }

    public ICharacterBuilder SetName(string name)
    {
        _character.Name = name;
        return this;
    }

    public ICharacterBuilder SetHeight(int height)
    {
        _character.Height = height;
        return this;
    }

    public ICharacterBuilder SetBuild(string build)
    {
        _character.Build = build;
        return this;
    }

    public ICharacterBuilder SetHairColor(string hairColor)
    {
        _character.HairColor = hairColor;
        return this;
    }

    public ICharacterBuilder SetEyeColor(string eyeColor)

```

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Пр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

```

    {
        _character.EyeColor = eyeColor;
        return this;
    }

    public ICharacterBuilder SetClothing(string clothing)
    {
        _character.Clothing = clothing;
        return this;
    }

    public ICharacterBuilder AddInventory(string inventory)
    {
        _character.Inventory.Add(inventory);
        return this;
    }

    public Character Build()
    {
        return _character;
    }
}

namespace Lab2.Task5.BUILDERS;
public class HeroBuilder : ICharacterBuilder
{
    private Character _character = new Character();

    public HeroBuilder()
    {
        Reset();
    }

    public void Reset()
    {
        _character = new Character();
        _character.Alignment = "Good";
    }

    public ICharacterBuilder SetName(string name)
    {
        _character.Name = name;
        return this;
    }

    public ICharacterBuilder SetHeight(int height)
    {
        _character.Height = height;
        return this;
    }

    public ICharacterBuilder SetBuild(string build)

```

		Геращенко Д. І.			ДУ «Житомирська політехніка».23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

```

    {
        _character.Build = build;
        return this;
    }

    public ICharacterBuilder SetHairColor(string hairColor)
    {
        _character.HairColor = hairColor;
        return this;
    }

    public ICharacterBuilder SetEyeColor(string eyeColor)
    {
        _character.EyeColor = eyeColor;
        return this;
    }

    public ICharacterBuilder SetClothing(string clothing)
    {
        _character.Clothing = clothing;
        return this;
    }

    public ICharacterBuilder AddInventory(string inventory)
    {
        _character.Inventory.Add(inventory);
        return this;
    }

    public Character Build()
    {
        return _character;
    }
}

using Lab2.Task5.BUILDERS;

namespace Lab2.Task5;

public class CharacterDirector
{
    public Character ConstructL(ICharacterBuilder builder)
    {
        return builder
            .SetName("Detective L")
            .SetHeight(179)
            .SetBuild("Weak")
            .SetHairColor("Brown")
            .SetEyeColor("Brown")
            .SetClothing("Rugs")
            .AddInventory("Sugar Cubes")
    }
}

```

		Геращенко Д. І.		
		Фаніт М. О.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

        .Build();
    }

    public Character ConstructYagami(ICharacterBuilder builder)
    {
        return builder
            .SetName("Light Yagami")
            .SetHeight(180)
            .SetBuild("Weak")
            .SetHairColor("Black")
            .SetEyeColor("Black")
            .SetClothing("Suit")
            .AddInventory("Death Note")
            .Build();
    }
}

using Lab2.Task5;
using Lab2.Task5.BUILDERS;
using Xunit;

namespace Lab2.Task5.Tests;

public class BuilderTests
{
    [Fact]
    public void HeroBuilder()
    {
        var director = new CharacterDirector();
        var builder = new HeroBuilder();
        Character hero = director.ConstructL(builder);

        Assert.Equal("Detective L", hero.Name);
        Assert.Equal(179, hero.Height);
        Assert.Equal("Weak", hero.Build);
        Assert.Equal("Brown", hero.HairColor);
        Assert.Equal("Brown", hero.EyeColor);
        Assert.Equal("Rugs", hero.Clothing);
        Assert.Equal(["Sugar Cubes"], hero.Inventory);
    }

    [Fact]
    public void EnemyBuilder()
    {
        var director = new CharacterDirector();
        var builder = new EnemyBuilder();
        Character enemy = director.ConstructYagami(builder);

        Assert.Equal("Light Yagami", enemy.Name);
        Assert.Equal(180, enemy.Height);
        Assert.Equal("Weak", enemy.Build);
    }
}

```

		Геращенко Д. І.		
		Фант М. О.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

        Assert.Equal("Black", enemy.HairColor);
        Assert.Equal("Black", enemy.EyeColor);
        Assert.Equal("Suit", enemy.Clothing);
        Assert.Equal(["Death Note"], enemy.Inventory);
    }

}

```

Результат виконання програми:

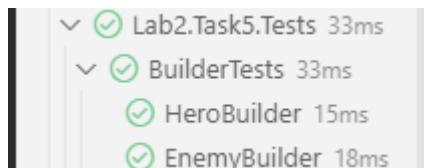


Рис 7 . Результат виконання завдання 5

Блок схема:

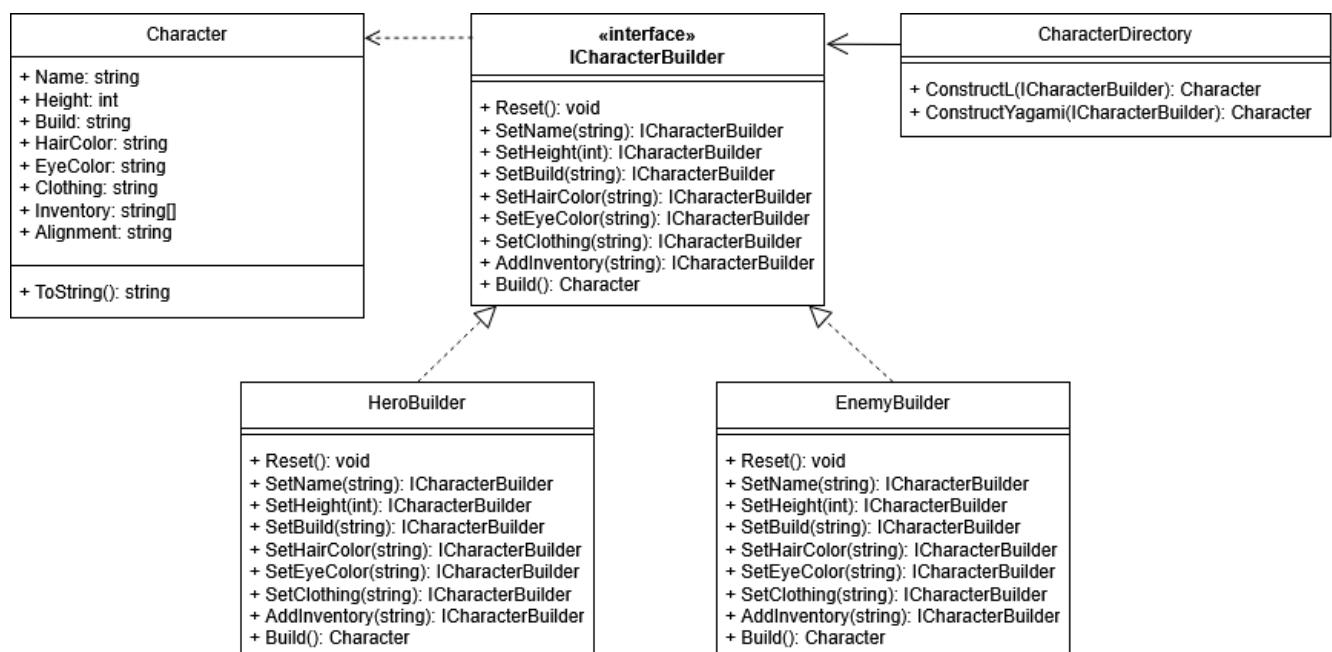


Рис 8. UML діаграма до завдання 5

Посилання на репозиторій GitHub:

<https://github.com/anyduck-forks/kpz/tree/main/Lab2>

Висновки: В ході лабораторної роботи ми сформували навички роботи із породжувальними шаблонами проєктування, такими як Фабричний метод, Фабрика, Абстрактна фабрика, Прототип, Одинак та Будівельник. Закріпили ці знання за допомогою виконання завдань на мові C#.

		Геращенко Д. І.			ДУ «Житомирська політехніка». 23.121.09.000 – Лр2	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		19